Bảng Hãng Xe

HX\_MaHang int primary key,

HX\_TenHangXe varchar(50),

HX\_DongXe varchar(30),

HX\_QuocGia varchar(30)

Bảng Loại Xe

LX\_MaLoai int primary key,

LX\_TenLoai varchar(50),

HX\_MaHang int (Khóa Ngoại)

Bảng Thông Số Kỹ Thuật

TSKT\_Ma int primary key,

TSKT\_DongCo varchar(50),

TSKT\_CongSuat int,

TSKT\_SoCho int

Bảng Xe

Xe\_Ma int primary key,

HX\_MaHang int (KN)

LX\_MaLoai int (KN),

TSKT\_Ma int (KN),

PhanKhucXe varchar(50),

GiaNiemYet float,

Xe\_PhienBan varchar(50)

Bảng Tỉnh Thành

Tinh\_Ma int primary key,

Tinh\_TenTinh varchar(50)

Bảng ChiPhi

CP\_Ma int primary key,

Xe\_Ma int  (KN),

CP\_Phisudungduongbo float,

CP\_PhiBaoHiemTrachNhiemDanSu float,

CP\_PhiDangKiem float

Bang HoaDon

HD\_Ma int primary key,

Xe\_Ma int (KN)

Tinh\_Ma int (KN),

CP\_Ma int (KN)

HD\_PhiDangKyBienSo float,

HD\_GiaLanBanh float

CREATE TABLE HangXe (

    HX\_MaHang INT PRIMARY KEY,

    HX\_TenHangXe VARCHAR(50),

    HX\_DongXe VARCHAR(30)

);

CREATE TABLE LoaiXe (

    LX\_MaLoai INT PRIMARY KEY,

    LX\_TenLoai VARCHAR(50),

    HX\_MaHang INT,

    FOREIGN KEY (HX\_MaHang) REFERENCES HangXe(HX\_MaHang)

);

CREATE TABLE ThongSoKyThuat (

    TSKT\_Ma INT PRIMARY KEY,

    TSKT\_DongCo VARCHAR(50),

    TSKT\_CongSuat INT,

    TSKT\_SoCho INT

);

CREATE TABLE Xe (

    Xe\_Ma INT PRIMARY KEY,

    HX\_MaHang INT,

    LX\_MaLoai INT,

    TSKT\_Ma INT,

    PhanKhucXe VARCHAR(50),

    GiaNiemYet FLOAT,

    Xe\_PhienBan VARCHAR(50),

    FOREIGN KEY (HX\_MaHang) REFERENCES HangXe(HX\_MaHang),

    FOREIGN KEY (LX\_MaLoai) REFERENCES LoaiXe(LX\_MaLoai),

    FOREIGN KEY (TSKT\_Ma) REFERENCES ThongSoKyThuat(TSKT\_Ma)

);

CREATE TABLE TinhThanh (

    Tinh\_Ma INT PRIMARY KEY,

    Tinh\_TenTinh VARCHAR(50)

);

CREATE TABLE ChiPhi (

    CP\_Ma INT PRIMARY KEY,

    Xe\_Ma INT,

    CP\_Phisudungduongbo FLOAT,

    CP\_PhiBaoHiemTrachNhiemDanSu FLOAT,

    CP\_PhiDangKiem FLOAT,

    FOREIGN KEY (Xe\_Ma) REFERENCES Xe(Xe\_Ma)

);

CREATE TABLE GiaLanBanh (

    GiaLanBanh\_Ma INT PRIMARY KEY,

    Xe\_Ma INT,

    Tinh\_Ma INT,

    PhiCuocBa FLOAT,

    PhiSuDungDuongBo FLOAT,

    PhiBaoHiem FLOAT,

    PhiDangKyBienSo FLOAT,

    PhiDangKiem FLOAT,

    FOREIGN KEY (Xe\_Ma) REFERENCES Xe(Xe\_Ma),

    FOREIGN KEY (Tinh\_Ma) REFERENCES TinhThanh(Tinh\_Ma)

);

-- Thêm dữ liệu cho bảng HangXe

INSERT INTO HangXe (HX\_MaHang, HX\_TenHangXe, HX\_DongXe) VALUES

(1, 'Toyota', 'Corolla'),

(2, 'Honda', 'Civic');

-- Thêm dữ liệu cho bảng LoaiXe

INSERT INTO LoaiXe (LX\_MaLoai, LX\_TenLoai, HX\_MaHang) VALUES

(1, 'Sedan', 1),

(2, 'Sedan', 2);

-- Thêm dữ liệu cho bảng ThongSoKyThuat

INSERT INTO ThongSoKyThuat (TSKT\_Ma, TSKT\_DongCo, TSKT\_CongSuat, TSKT\_SoCho) VALUES

(1, '1.5L', 150, 5),

(2, '1.6L', 160, 5);

-- Thêm dữ liệu cho bảng Xe

INSERT INTO Xe (Xe\_Ma, HX\_MaHang, LX\_MaLoai, TSKT\_Ma, PhanKhucXe, GiaNiemYet, Xe\_PhienBan) VALUES

(1, 1, 1, 1, 'Phân khúc A', 20000.0, '2023'),

(2, 2, 2, 2, 'Phân khúc B', 22000.0, '2023');

-- Thêm dữ liệu cho bảng TinhThanh

INSERT INTO TinhThanh (Tinh\_Ma, Tinh\_TenTinh) VALUES

(1, 'Cần Thơ'),

(2, 'Hồ Chí Minh');

-- Thêm dữ liệu cho bảng ChiPhi

INSERT INTO ChiPhi (CP\_Ma, Xe\_Ma, CP\_Phisudungduongbo, CP\_PhiBaoHiemTrachNhiemDanSu, CP\_PhiDangKiem) VALUES

(1, 1, 500.0, 200.0, 100.0),

(2, 2, 550.0, 220.0, 110.0);

-- Thêm dữ liệu cho bảng GiaLanBanh

INSERT INTO GiaLanBanh (GiaLanBanh\_Ma, Xe\_Ma, Tinh\_Ma, PhiCuocBa, PhiSuDungDuongBo, PhiBaoHiem, PhiDangKyBienSo, PhiDangKiem) VALUES

(1, 1, 1, 10, 500, 200, 100, 100),

(2, 2, 2, 10, 550, 220, 110, 110);

### **1. Viết thủ tục cho phép nhập vào 1 xe mới:**

CREATE OR REPLACE PROCEDURE InsertNewCar(

    p\_HX\_TenHangXe IN VARCHAR2,

    p\_HX\_DongXe IN VARCHAR2,

    p\_Xe\_PhienBan IN VARCHAR2,

    p\_PhanKhucXe IN VARCHAR2,

    p\_TSKT\_DongCo IN VARCHAR2,

    p\_GiaNiemYet IN FLOAT

)

IS

    v\_HX\_MaHang INT;

    v\_Xe\_Ma INT;

BEGIN

    -- Thêm hãng xe nếu chưa tồn tại

    INSERT INTO HangXe (HX\_TenHangXe, HX\_DongXe)

    VALUES (p\_HX\_TenHangXe, p\_HX\_DongXe)

    RETURNING HX\_MaHang INTO v\_HX\_MaHang;

    -- Thêm xe nếu chưa tồn tại

    INSERT INTO Xe (HX\_MaHang, Xe\_PhienBan, PhanKhucXe, TSKT\_DongCo, GiaNiemYet)

    VALUES (v\_HX\_MaHang, p\_Xe\_PhienBan, p\_PhanKhucXe, p\_TSKT\_DongCo, p\_GiaNiemYet)

    RETURNING Xe\_Ma INTO v\_Xe\_Ma;

    COMMIT;

END;

### **2. Gọi thủ tục để Nhập liệu như bảng trên:**

BEGIN

    -- Gọi thủ tục InsertNewCar để nhập thông tin về xe mới

    InsertNewCar('Toyota', 'Corolla', '2024', 'Phân khúc A', '1.5L', 25000.0);

END;

3. Viết thủ tục cho phép cập nhật giá niêm yết của 1 xe nào đó.

CREATE OR REPLACE PROCEDURE IncreaseCarPrice(

    p\_Xe\_Ma IN INT

)

IS

BEGIN

    -- Cập nhật giá niêm yết mới của xe

    UPDATE Xe

    SET GiaNiemYet = GiaNiemYet + 10000000

    WHERE Xe\_Ma = p\_Xe\_Ma;

    COMMIT;

END;

EXECUTE IncreaseCarPrice(1);

5. Viết hàm trả về giá của 1 xe nào đó

CREATE OR REPLACE FUNCTION GetCarPrice(

    p\_Xe\_Ma IN INT

)

RETURN FLOAT

IS

    v\_Price FLOAT;

BEGIN

    -- Lấy giá niêm yết của xe với mã tương ứng

    SELECT GiaNiemYet INTO v\_Price

    FROM Xe

    WHERE Xe\_Ma = p\_Xe\_Ma;

    RETURN v\_Price;

END;

DECLARE

    v\_CarPrice FLOAT;

BEGIN

    EXECUTE :v\_CarPrice := GetCarPrice(1); -- Gán giá trị trả về của hàm cho biến v\_CarPrice

    DBMS\_OUTPUT.PUT\_LINE('Giá của xe là: ' || v\_CarPrice);

END;

6. Viết hàm trả về loại xe, hãng xe có giá cao nhất

CREATE OR REPLACE FUNCTION GetHighestPriceCarTypeAndBrand

RETURN VARCHAR2

IS

    v\_Result VARCHAR2(100); -- Kết quả trả về (loại và hãng xe có giá cao nhất)

BEGIN

    -- Sử dụng câu lệnh SELECT để lấy loại và hãng xe có giá cao nhất

    SELECT LX\_TenLoai || ' - ' || HX\_TenHangXe

    INTO v\_Result

    FROM Xe

    JOIN LoaiXe ON Xe.LX\_MaLoai = LoaiXe.LX\_MaLoai

    JOIN HangXe ON Xe.HX\_MaHang = HangXe.HX\_MaHang

    ORDER BY GiaNiemYet DESC

    FETCH FIRST 1 ROW ONLY;

    RETURN v\_Result;

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN NULL; -- Trả về NULL nếu không tìm thấy dữ liệu

    WHEN OTHERS THEN

        RETURN NULL; -- Xử lý các lỗi khác và trả về NULL

END;

DECLARE

    v\_HighestPriceCarTypeAndBrand VARCHAR2(100);

BEGIN

    v\_HighestPriceCarTypeAndBrand := GetHighestPriceCarTypeAndBrand();

    DBMS\_OUTPUT.PUT\_LINE('Loại xe và hãng xe có giá cao nhất là: ' || v\_HighestPriceCarTypeAndBrand);

END;

7. Viết thủ tục để xóa 1 xe nào đó

CREATE OR REPLACE PROCEDURE DeleteCar(

    p\_Xe\_Ma IN INT

)

IS

BEGIN

    -- Xóa xe dựa trên mã

    DELETE FROM Xe

    WHERE Xe\_Ma = p\_Xe\_Ma;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Xóa xe có mã ' || p\_Xe\_Ma || ' thành công.');

EXCEPTION

    WHEN OTHERS THEN

        ROLLBACK;

        DBMS\_OUTPUT.PUT\_LINE('Đã xảy ra lỗi khi xóa xe: ' || SQLERRM);

END;

//xuat ra

BEGIN

    DeleteCar(1);

END;

8. Viết hàm trả về giá lăn bánh ở Cần Thơ của 1 xe nào đó, biết rằng giá lăn bánh được tính theo công thức sau:

CREATE OR REPLACE FUNCTION CalculateRollingPriceInCanTho(

    p\_Xe\_Ma IN INT

)

RETURN FLOAT

IS

    v\_GiaNiemYet FLOAT;

    v\_CP\_PhiCuocBa FLOAT := 0.1; -- Phí cước bạ 10%

    v\_CP\_PhiSDDuongBo FLOAT := 1000000; -- Giả sử phí sử dụng đường bộ là 1,000,000 VND/năm

    v\_CP\_PhiBHTrachNhiem FLOAT := 500000; -- Giả sử phí bảo hiểm trách nhiệm nhân sự là 500,000 VND/năm

    v\_CP\_PhiDangKyBienSo FLOAT := 200000; -- Giả sử phí đăng ký biển số là 200,000 VND

    v\_CP\_PhiDangKiem FLOAT := 300000; -- Giả sử phí đăng kiểm là 300,000 VND

    v\_RollingPrice FLOAT;

BEGIN

    -- Lấy giá niêm yết của xe

    SELECT GiaNiemYet INTO v\_GiaNiemYet

    FROM Xe

    WHERE Xe\_Ma = p\_Xe\_Ma;

    -- Tính giá lăn bánh tại Cần Thơ theo công thức

    v\_RollingPrice := v\_GiaNiemYet

                    + (v\_GiaNiemYet \* v\_CP\_PhiCuocBa)

                    + v\_CP\_PhiSDDuongBo

                    + v\_CP\_PhiBHTrachNhiem

                    + v\_CP\_PhiDangKyBienSo

                    + v\_CP\_PhiDangKiem;

    RETURN v\_RollingPrice;

END;

DECLARE

    v\_RollingPrice FLOAT;

BEGIN

    v\_RollingPrice := CalculateRollingPriceInCanTho(1); -- Tính giá lăn bánh của xe có mã 1

    DBMS\_OUTPUT.PUT\_LINE('Giá lăn bánh ở Cần Thơ của xe là: ' || v\_RollingPrice || ' VND');

END;

9. Viết trigger để theo dõi việc sửa giá trong bảng xe

CREATE TABLE LichSuCapNhatGiaXe (

    Xe\_Ma INT,

    HX\_TenHangXe VARCHAR(50),

    GiaCu FLOAT,

    GiaMoi FLOAT,

    NgaySua DATE

);

CREATE OR REPLACE TRIGGER TrackPriceChanges

AFTER UPDATE OF GiaNiemYet ON Xe

FOR EACH ROW

DECLARE

    v\_HX\_TenHangXe VARCHAR2(50);

BEGIN

    -- Lấy tên hãng xe tương ứng với mã hãng xe

    SELECT HX\_TenHangXe INTO v\_HX\_TenHangXe

    FROM HangXe

    WHERE HX\_MaHang = :NEW.HX\_MaHang;

    -- Ghi lại thông tin về việc cập nhật giá vào bảng LichSuCapNhatGiaXe

    INSERT INTO LichSuCapNhatGiaXe (Xe\_Ma, HX\_TenHangXe, GiaCu, GiaMoi, NgaySua)

    VALUES (:NEW.Xe\_Ma, v\_HX\_TenHangXe, :OLD.GiaNiemYet, :NEW.GiaNiemYet, SYSDATE);

END;

SELECT \* FROM LichSuCapNhatGiaXe;

10. Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe

CREATE TABLE LichSuXoaXe (

    Xe\_Ma INT,

    HX\_TenHangXe VARCHAR(50),

    LX\_TenLoai VARCHAR(50),

    TSKT\_DongCo VARCHAR(50),

    PhanKhucXe VARCHAR(50),

    GiaNiemYet FLOAT,

    Xe\_PhienBan VARCHAR(50),

    ThoiGianXoa DATE

);

CREATE OR REPLACE TRIGGER TrackDeleteXe

BEFORE DELETE ON Xe

FOR EACH ROW

BEGIN

    INSERT INTO LichSuXoaXe (Xe\_Ma, HX\_TenHangXe, LX\_TenLoai, TSKT\_DongCo, PhanKhucXe, GiaNiemYet, Xe\_PhienBan, ThoiGianXoa)

    VALUES (:OLD.Xe\_Ma,

            (SELECT HX\_TenHangXe FROM HangXe WHERE HX\_MaHang = :OLD.HX\_MaHang),

            (SELECT LX\_TenLoai FROM LoaiXe WHERE LX\_MaLoai = :OLD.LX\_MaLoai),

            (SELECT TSKT\_DongCo FROM ThongSoKyThuat WHERE TSKT\_Ma = :OLD.TSKT\_Ma),

            :OLD.PhanKhucXe, :OLD.GiaNiemYet, :OLD.Xe\_PhienBan, SYSDATE);

END;

SELECT \* FROM LichSuXoaXe

11. Tạo các người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..)

**-- Tạo người dùng QuanLy**

**CREATE USER QuanLy IDENTIFIED BY QuanLyPass;**

**-- Cấp quyền quản trị cho người dùng QuanLy**

**GRANT CONNECT, RESOURCE, DBA TO QuanLy;**

**-- Tạo người dùng NhanVienBanXe**

**CREATE USER NhanVienBanXe IDENTIFIED BY NhanVienPass;**

**-- Cấp quyền chỉ đọc cho người dùng NhanVienBanXe**

**GRANT CONNECT, SELECT ON <ten\_bang> TO NhanVienBanXe; -- Thay <ten\_bang> bằng tên bảng bạn muốn cấp quyền**

**-- Tạo người dùng khác và cấp quyền tương ứng nếu cần**

**Câu lệnh** GRANT SELECT ON <ten\_bang> TO NhanVienBanXe **cấp quyền chỉ đọc cho người dùng** NhanVienBanXe **trên một bảng cụ thể.**

bài 2

**Quản Lý Tiệm bán Bánh**

CREATE TABLE  SanPham (

    MaSanPham  VARCHAR (15) PRIMARY KEY,

    TenSanPham  VARCHAR(30) NOT NULL,

    Gia FLOAT check(Gia >0),

    SoLuongCoSan INT NOT NULL

);

CREATE TABLE  Khach\_Hang (

    MaKhachHang  VARCHAR (15)PRIMARY KEY,

    TenKhachHang VARCHAR(30) NOT NULL,

    Email VARCHAR(30) NOT NULL,

    SoDienThoai VARCHAR(20) NOT NULL,

    DiaChi VARCHAR(30) NOT NULL

);

CREATE TABLE  DonHang(

   MaDonHang VARCHAR (15) PRIMARY KEY,

    MaKhachHang VARCHAR (15),

    NgayDatHang DATE NOT NULL,

    TongSoTien FLOAT check(TongSoTien >0),

    FOREIGN KEY (MaKhachHang) REFERENCES Khach\_Hang (MaKhachHang)

);

CREATE TABLE ChiTietDonHang (

    MaChiTiet  VARCHAR(15) PRIMARY KEY,

    MaDonHang VARCHAR(15),

    MaSanPham VARCHAR(15),

    SoLuong INT NOT NULL check(SoLuong >0),

    ThanhTien FLOAT check(ThanhTien >0),

    FOREIGN KEY (MaDonHang ) REFERENCES DonHang(MaDonHang ),

    FOREIGN KEY (MaSanPham ) REFERENCES SanPham (MaSanPham )

);

CREATE TABLE Thanh\_Toan (

    MaThanhToan VARCHAR(15) PRIMARY KEY,

    DonDatHang VARCHAR(15) NOT NULL,

    NgayThanhToan DATE NOT NULL,

    MaDonHang VARCHAR(15) ,

    SoTienThanhToan FLOAT check(SoTienThanhToan  >0),

    FOREIGN KEY (MaDonHang ) REFERENCES DonHang(MaDonHang )

);

CREATE TABLE NhanVien(

    MaNhanVien VARCHAR(15) PRIMARY KEY,

    TenNhanVien VARCHAR(30) NOT NULL,

    Luong FLOAT check(Luong >0)

);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP001', 'Bánh mì socola', 5000, 7);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP002', 'Bánh bơ tỏi', 7000, 9);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP003', 'Bánh kem dâu', 6000, 20);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP004', 'Bánh ngọt hạt điều', 10000, 40);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP005', 'Bánh trứng muối', 12000, 25);

-- Nhập dữ liệu cho bảng Khach\_Hang

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH001', 'Nguyễn Văn A', 'nguyenvana@gmail.com', '0987654321', 'TP.HCM');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH002', 'Trần Thị B', 'tranthib@gmail.com', '0912345678', 'TP.Cần Thơ');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH003', 'Lê Thị C', 'lethic@gmail.com', '0978123456', 'Hà Nội');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH004', 'Phạm Văn D', 'phamvand@gmail.com', '0909123456', 'Huế');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH005', 'Hoàng Văn E', 'hoangvane@gmail.com', '0988777666', 'Vĩnh Long');

-- Nhập dữ liệu cho bảng DonHang

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH001', 'KH001', to\_date('2024-03-20','yyyy-mm-dd'), 25000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH002', 'KH002', to\_date('2024-03-20','yyyy-mm-dd'), 27000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH003', 'KH003', to\_date('2024-03-20','yyyy-mm-dd'), 18000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH004', 'KH004', to\_date('2024-03-20','yyyy-mm-dd'), 15000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH005', 'KH005', to\_date('2024-03-20','yyyy-mm-dd'), 22000);

-- Nhập dữ liệu cho bảng ChiTietDonHang

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT001', 'DH001', 'SP001', 5, 25000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT002', 'DH002', 'SP002', 3, 21000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT003', 'DH002', 'SP004', 2, 20000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT004', 'DH003', 'SP003', 4, 24000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT005', 'DH005', 'SP005', 2, 24000);

-- Nhập dữ liệu cho bảng Thanh\_Toan

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT001', 'DH001', to\_date('2024-03-20','yyyy-mm-dd'), 'DH001', 25000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT002', 'DH002', to\_date('2024-03-20','yyyy-mm-dd'), 'DH002', 27000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT003', 'DH003', to\_date('2024-03-20','yyyy-mm-dd'), 'DH003', 18000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT004', 'DH004', to\_date('2024-03-20','yyyy-mm-dd'), 'DH004', 15000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT005', 'DH005', to\_date('2024-03-20','yyyy-mm-dd'), 'DH005', 22000);

-- Nhập dữ liệu cho bảng NhanVien

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV001', 'Nguyễn Văn Nam', 10000000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV002', 'Trần Thị Hương', 9000000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV003', 'Lê Văn Đức', 8500000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV004', 'Phạm Thị Lan', 9500000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV005', 'Hoàng Văn Long', 10500000);

**Hàm Tính tổng số lượng tồn kho của tất cả sản phẩm.**

CREATE OR REPLACE FUNCTION TinhTongTonKho RETURN NUMBER

IS

    TonKho NUMBER := 0;

BEGIN

    FOR SanPham\_L IN (SELECT SoLuongCoSan FROM SanPham) LOOP

        TonKho := TonKho + SanPham\_L.SoLuongCoSan;

    END LOOP;

    RETURN TonKho ;

END;

-- xem kết quả

SELECT TinhTongTonKho from dual;

**2. Hàm kiểm tra sản phẩm có tồn kho hay không.**

CREATE OR REPLACE FUNCTION KiemTraTonKho(maSP IN VARCHAR) RETURN VARCHAR

IS

    soLuong NUMBER;

    result VARCHAR(20);

BEGIN

    -- Lấy số lượng tồn kho của sản phẩm với mã sản phẩm được cung cấp

    SELECT SoLuongCoSan INTO soLuong

    FROM SanPham

    WHERE MaSanPham = maSP;

    -- Kiểm tra nếu số lượng tồn kho lớn hơn 0, trả về tồn tại, ngược lại trả về không tồn tại

    IF soLuong > 0 THEN

        result := 'Tồn tại';

    ELSE

        result := 'Không tồn tại';

    END IF;

    RETURN result;

END ;

-- xem kết quả

SELECT KiemTraTonKho('SP001') as kiemtrakho FROM dual;

**3. Hàm tính tổng lương của nhân viên.**

CREATE OR REPLACE FUNCTION TongLuongNhanVien  RETURN NUMBER

 IS

    TongTienLuong NUMBER := 0;

BEGIN

      FOR NhanVien\_L IN (SELECT Luong FROM NhanVien)

      LOOP

        TongTienLuong := TongTienLuong + NhanVien\_L.Luong;

    END LOOP;

    RETURN TongTienLuong;

END;

-- chạy đoạn này để xem kết quả

SELECT TongLuongNhanVien FROM DUAL;

**4. Thủ tục thêm 1 sản phẩm mới vào bảng sản phẩm.**

CREATE OR REPLACE PROCEDURE ThemSanPhamMoi(

    Masp VARCHAR ,

    Tensp VARCHAR,

    SP\_Gia FLOAT,

    SP\_SoLuongCoSan INT

) AS

BEGIN

    INSERT INTO SanPham (MaSanPham, TenSanPham, Gia, SoLuongCoSan)

    VALUES (Masp , Tensp , SP\_Gia , SP\_SoLuongCoSan);

    COMMIT;

END;

-- thêm sp mới

EXECUTE ThemSanPhamMoi('SP006', 'bánh macaron', 13000, 30);

-- gọi lại bảng để xem kết quả

SELECT \* FROM SanPham;

**5. Thủ tục xóa nhân viên theo mã nhân viên.**

CREATE OR REPLACE PROCEDURE Xoanhanvien(Ma\_NhanVien Varchar)

IS

BEGIN

     DELETE FROM NhanVien

     WHERE MaNhanVien = Ma\_NhanVien;

     COMMIT;

END Xoanhanvien;

-- gọi thủ tục xóa

EXECUTE Xoanhanvien('NV002');

-- xem kết quả

SELECT \* FROM NhanVien;

**6. Thủ tục tăng lương thêm 500000 cho nhân viên nào lương dưới 10 triệu.**

CREATE OR REPLACE PROCEDURE TangLuongChoNhanVien(tang number) AS

BEGIN

    -- Thêm 500000 cho nhân viên có mức lương dưới 10 triệu

    UPDATE NhanVien

    SET Luong = Luong + tang

    WHERE Luong < 10000000;

END TangLuongChoNhanVien;

-- gọi thủ tục xóa

EXECUTE TangLuongChoNhanVien (500000);

-- xem kết quả

SELECT \*FROM NhanVien;

**7. Viết trigger cập nhật thay đổi lương của nhân viên.**

CREATE TABLE TheoDoiCapNhatLuong (

    MaNhanVien VARCHAR(10),

    LuongCu FLOAT,

    LuongMoi FLOAT,

    ThoiGianCapNhat TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiCapNhatLuongNhanVien

AFTER UPDATE OF Luong ON NhanVien

FOR EACH ROW

BEGIN

    IF UPDATING('Luong') THEN

        INSERT INTO TheoDoiCapNhatLuong (MaNhanVien, LuongCu, LuongMoi, ThoiGianCapNhat)

        VALUES (:OLD.MaNhanVien, :OLD.Luong, :NEW.Luong, SYSTIMESTAMP);

    END IF;

END;

-- Thêm một nhân viên mới vào bảng NhanVien

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong)

VALUES ('NV007', 'Trần Văn A', 15000000);

-- Cập nhật lương cho nhân viên vừa thêm để kích hoạt trigger

UPDATE NhanVien SET Luong = 16000000 WHERE MaNhanVien = 'NV007';

-- xem kết quả

SELECT \* FROM TheoDoiCapNhatLuong

**8. Viết trigger cập nhật nhân viên bị xóa khỏi bảng nhân viên.**

CREATE TABLE Xoa\_NhanVien(

    MaNhanVien VARCHAR(15),

    HoTen VARCHAR(30),

    ThoiGianXoa TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiXoaNhanVien

AFTER DELETE ON NhanVien

FOR EACH ROW

DECLARE

    v\_ThoiGianXoa TIMESTAMP ;

BEGIN

    -- Lấy thời gian hiện tại

    v\_ThoiGianXoa := SYSTIMESTAMP;

    -- Ghi nhận thông tin về nhân viên bị xóa vào bảng XoaNhanVien

    INSERT INTO Xoa\_NhanVien (MaNhanVien, HoTen, ThoiGianXoa)

    VALUES (:OLD.MaNhanVien, :OLD. TenNhanVien, v\_ThoiGianXoa);

END Xoa\_NhanVien;

-- thêm nhân viên vào

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV006', 'Nguyễn Văn Nam', 12000000);

-- xóa nhân viên

DELETE FROM NhanVien WHERE MaNhanVien = 'NV006';

-- xem kết quả nhân viên bị xóa

SELECT \* FROM Xoa\_NhanVien

**9. Viết trigger để lưu lại thời gian thêm sản phẩm mới vào.**

CREATE TABLE them\_banhmoi (

    Mabanh VARCHAR(15),

    Tenbanh VARCHAR(30),

    Giabanh FLOAT,

    SoLuongbanh INT,

    ThoiGianthem TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiThemBanh

AFTER INSERT ON SanPham

FOR EACH ROW

DECLARE

    v\_ThoiGianThem TIMESTAMP;

BEGIN

    -- Lấy thời gian hiện tại

    v\_ThoiGianThem := SYSTIMESTAMP;

    -- Ghi nhận thông tin về bánh mới được thêm vào

    INSERT INTO them\_banhmoi (Mabanh, Tenbanh, Giabanh, SoLuongbanh, ThoiGianthem)

    VALUES (:NEW.MaSanPham, :NEW.TenSanPham, :NEW.Gia, :NEW.SoLuongCoSan, v\_ThoiGianThem);

END;

-- thêm sản phẩm mới vào

INSERT INTO SanPham (MaSanPham, TenSanPham, Gia, SoLuongCoSan)

VALUES ('SP007', 'Bánh mỳ', 20000, 30);

-- Xem kết quả

SELECT \* FROM them\_banhmoi;

bài 3:

**Bài tập 3**

Tóm tắt câu trả lời cho các vấn đề dưới đây (Không copy & paste):

HQTCSDL là gì? Các HQTCSDL hiện nay

HQTCSDL là một hệ thống phần mềm cho phép người dùng định nghĩa, tạo và duy trì CSDL đồng thời cung cấp dịch vụ truy cập đến CSDL này một cách có quản lý.

Các HQTCSDL hiện nay bao gồm: Oracle, DB2, MS SQL Server, MS Access và các phần mềm mở như MySQL, PostGreSQL,...

Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì?

HQTCSDL

 Người dùng có thể lưu trữ, truy xuất và cập nhật dữ liệu thông qua việc sử dụng các ngôn ngữ định nghĩa.

Bảo vệ CSDL khỏi những đe dọa có chủ ý hay vô tình thông qua các biện pháp có sử dụng máy tính hoặc không có sử dụng máy tính.

Có thể phục hồi từ toàn bộ cơ sở dữ liệu hoặc các bảng cụ thể, tùy thuộc vào nhu cầu và cấu trúc backup.

Nếu có nhiều người dùng cùng lúc thì có khả năng xảy ra tình huống cập nhật làm cho dữ liệu mất tính nhất quán.Thường có các cơ chế kiểm soát đồng thời (concurrency control) để đảm bảo tính nhất quán của dữ liệu khi nhiều người cùng thao tác.

Excel

Cung cấp các tính năng cơ bản về bảo mật như mật khẩu bảo vệ tệp Excel, nhưng không có nhiều tùy chọn bảo mật phức tạp như HQTCSDL.

Thường phục hồi từ toàn bộ tập tin hoặc một phần của nó, không có khả năng phục hồi từng phần riêng lẻ của dữ liệu.

Excel không có tính năng kiểm soát đồng thời như HQTCSDL. Nếu nhiều người truy cập cùng một tệp Excel, có thể xảy ra xung đột dữ liệu hoặc mất tính nhất quán.

DBA là ai? Nhiệm vụ?

DBA là viết tắt của Database Administrator, đó là một vị trí quan trọng trong các tổ chức có cơ sở dữ liệu. Một DBA có trách nhiệm quản lý, duy trì và bảo vệ.

Nhiệm vụ: Quản lý CSDL, bảo mật dữ liệu, sao lưu và phục hồi, hỗ trợ và giám sát hiệu suất và điều chỉnh hệ thống.

Các phương pháp bảo vệ dữ liệu

An toàn trong CSDL là bảo vệ CSDL khỏi những đe dọa có chủ ý hay vô tình.

Cấp quyền là sự gán quyền cho một người dùng hay chương trình để có thể truy cập vào một hệ thống hay một đối tượng của hệ thống.

Khung nhìn thay vì cấp quyền cho người dùng trên nhiều bảng, ta có thể gán các quyền thích hợp trên một khung nhìn định nghĩa trên bảng này.

Sao lưu và phục hồi là quá trình chép CSDL và các tập tin nhật ký vào các thiết bị lưu trữ dự phòng một cách định kỳ.

Toàn vẹn dữ liệu là đảm bảo rằng dữ liệu không bị thay đổi, biến đổi hoặc mất mát một cách không được ủy quyền hay không được mong muốn.

Mật hóa dữ liệu là sự mã hóa dữ liệu bằng một giải thuật đặc biệt làm cho dữ liệu không thể đọc được nếu không có khóa giải mã.

RAID là một công nghệ được sử dụng để tăng cường hiệu suất và độ tin cậy của hệ thống lưu trữ dữ liệu bằng cách kết hợp nhiều ổ đĩa vật lý thành một đơn vị lưu trữ logic.

Giao dịch là gì? Tại sao phải cần GD?

Giao dịch là một hành động hay một chuỗi các hành động được thực hiện bởi một người dùng hoặc một chương trình ứng dụng.

Tại sao cần giao dịch? vì nó giúp đảm bảo tính nhất quán và độ tin cậy của hệ thống thông tin.

Các trạng thái của GD

Hoạt động (Active) Trạng thái khởi đầu; giao dịch giữ trong trạng thái

này trong khi nó đang thực hiện.

Hoàn tất một phần (Partially Committed) Sau khi lệnh cuối cùng

được thực hiện.

Thất bại (Failed) Sau khi phát hiện rằng sự thực hiện không thể tiếp tục

được nữa.

Hủy bỏ (Aborted) Sau khi giao dịch đã bị cuộn lại (rolled back) và

CSDL đã phục hồi lại trạng thái của nó trước khi khởi động giao dịch.

Thuộc tính ACID là gì?

Thuộc tính ACID là một tập hợp các đặc điểm quan trọng để đảm bảo tính nhất quán và độ tin cậy của giao dịch trong cơ sở dữ liệu.

Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục?

Xung đột dữ liệu: Khi nhiều người dùng cố gắng truy cập hoặc chỉnh sửa cùng một dữ liệu đồng thời, có thể xảy ra xung đột dữ liệu khi dữ liệu được thay đổi một cách không đồng nhất, gây ra sự không nhất quán hoặc mất mát dữ liệu.

Thất bại của giao dịch: Trong một môi trường đa người dùng, giao dịch của một người dùng có thể bị ảnh hưởng bởi các giao dịch khác đang diễn ra, dẫn đến thất bại hoặc không hoàn thành giao dịch một cách thành công.

Hiệu suất giảm sút: Sự cạnh tranh giữa các người dùng có thể dẫn đến tình trạng mất hiệu suất khi nhiều người cùng truy cập và thao tác trên cùng một hệ thống hoặc tài nguyên cùng một lúc.

Giải pháp:

Kiểm soát đồng thời (Concurrency Control): Sử dụng các kỹ thuật kiểm soát đồng thời như khóa, giao thức đọc ghi, và kiểm soát đồng thời để đảm bảo rằng các thay đổi dữ liệu được thực hiện một cách an toàn và nhất quán.

Quản lý giao dịch (Transaction Management): Sử dụng các cơ chế quản lý giao dịch để đảm bảo tính toàn vẹn và đồng nhất của giao dịch, bao gồm các nguyên tắc ACID.

Phân chia tài nguyên (Resource Partitioning): Phân chia tài nguyên như cơ sở dữ liệu hoặc máy chủ thành các phần nhỏ hơn để giảm thiểu sự cạnh tranh và tăng cường hiệu suất.

Lịch trình là gì? Lịch trình tuần tự và không tuần tự

Lịch trình là một chuỗi các thao tác thực hiện bởi một tập hợp các GD cạnh tranh mà vẫn đảm bảo thứ tự của các thao tác trong từng GD đơn lẻ.

tuần tự (serial schedule) là một lịch trình trong đó các thao tác của một GD được thực hiện liên tiếp nhau, không có bất kỳ thao tác nào của các GD khác xen vào giữa.

trình không tuần tự (nonserial schedule) là một lịch trình trong đó các thao tác từ một tập hợp các giao dịch cạnh tranh đan xen lẫn nhau.

Các kỹ thuật quản lý cạnh tranh: khóa chốt và nhãn thời gian.

Bi quan (lock): là các tiếp cận thận trọng vì các phương pháp này làm cho các GD bị ngưng trệ khi chúng xảy ra xung đột với các giao dịch khác tại một thời điểm nào đó trong tương lai.

Sử dụng Giao thức khóa 2 kỳ để giải quyết cạnh tranh

Các vấn đề khi sử dụng 2PL

..

Deadlock

- Khóa chết là một tình huống bế tắc khi hai hay nhiều GD đang chờ lẫn nhau để có được các khóa đang giữ bởi đối phương.

Xử lý deadlock

- Chỉ có một cách để phá bỏ tình trạng khóa chết là hủy bỏ 1 hoặc nhiều GD

Sử dụng nhãn thời gian

- Các nhãn thời gian có thể được tạo ra một cách đơn giản bằng cách sử dụng đồng hồ của hệ thống ở thời điểm GD bắt đầu hoặc bằng cách tăng một con số đếm luận lý mỗi khi một GD mới bắt đầu.

Lạc quan: 3 kỳ (đọc, kiểm tra, ghi)

Độ mịn của mục dữ liệu

- Độ mịn là kích cỡ của mục dữ liệu được chọn như làm một đơn vị bảo vệ bởi giao thức điều khiển cạnh tranh.

Phục hồi dữ liệu

Tại sao cần phục hồi DL

- Phục hồi dữ liệu là quá trình khôi phục lại thông tin từ các hệ thống lưu trữ khi dữ liệu gặp sự cố hoặc bị mất. Đây là một khía cạnh quan trọng của quản lý dữ liệu và bảo mật thông tin, vì nó đảm bảo rằng thông tin quan trọng không bị mất hoặc bị hỏng.

Điểm kiểm tra là gì (check point)

- check point là một thao tác khá rẻ, và nó thường được thực hiện khoảng 4 lần trong 1 giờ, Bằng cách này, ta sẽ không phải phục hồi nhiều hơn lượng công việc trong khoảng 15-20 phút.

Các kỹ thuật phục hồi DL

Cập nhật trì hoãn.

Cập nhật tức thì.

Tạo trang bóng.

bài 1

CREATE TABLE HangXe (

    MaHang VARCHAR (15) PRIMARY KEY,

    TenHangXe VARCHAR(50)

);

CREATE TABLE DongXe (

    MaDong VARCHAR (15) PRIMARY KEY,

    TenDong VARCHAR(50),

    MaHang VARCHAR (15),

    FOREIGN KEY (MaHang) REFERENCES HangXe(MaHang)

);

CREATE TABLE DongCo (

    MaDongCo VARCHAR (15) PRIMARY KEY,

    TenDongCo VARCHAR(50)

);

CREATE TABLE PhanKhucXe (

    MaPhanKhuc VARCHAR (15) PRIMARY KEY,

    TenPhanKhuc VARCHAR(50),

    MaDongCo VARCHAR (15),

    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)

);

CREATE TABLE PhienBan (

    MaPhienBan VARCHAR (15) PRIMARY KEY,

    TenPhienBan VARCHAR(50),

    MaPhanKhuc VARCHAR (15),

    FOREIGN KEY (MaPhanKhuc) REFERENCES PhanKhucXe(MaPhanKhuc)

);

CREATE TABLE GiaPhienBan (

    MaGiaPhienBan VARCHAR(15) PRIMARY KEY,

    MaPhienBan VARCHAR(15),

    GiaNiemYet FLOAT,

    FOREIGN KEY (MaPhienBan) REFERENCES PhienBan(MaPhienBan)

);

CREATE TABLE NoiDangKy (

    MaTinhThanh VARCHAR(15)PRIMARY KEY,

    TenTinhThanh VARCHAR(50),

    PhiDangKy FLOAT

);

CREATE TABLE ThongSoKyThuat (

    TSKT\_Ma VARCHAR(15) PRIMARY KEY,

    MaDongCo VARCHAR (15) ,

    CongSuat INT,

    SoCho INT,

    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)

);

CREATE TABLE Xe (

    MaXe  VARCHAR(15) PRIMARY KEY,

    HangXe\_Ma VARCHAR(15),

    DongXe\_Ma VARCHAR(15),

    PhanKhucXe\_Ma VARCHAR(15),

    TSKT\_Ma VARCHAR(15),

    MaTinhThanh  VARCHAR(15),

    FOREIGN KEY (HangXe\_Ma) REFERENCES HangXe(MaHang),

    FOREIGN KEY (DongXe\_Ma) REFERENCES DongXe(MaDong),

    FOREIGN KEY (PhanKhucXe\_Ma) REFERENCES PhanKhucXe(MaPhanKhuc),

    FOREIGN KEY (TSKT\_Ma) REFERENCES ThongSoKyThuat(TSKT\_Ma),

    FOREIGN KEY (MaTinhThanh ) REFERENCES NoiDangKy(MaTinhThanh )

);

CREATE TABLE GiaXe (

    HangXe\_Ma VARCHAR(15),

    DongXe\_Ma VARCHAR(15),

    PhienBan\_Ma VARCHAR(15),

    PhanKhucXe\_Ma VARCHAR(15),

    DongCo\_Ma VARCHAR(15),

    GiaNiemYet FLOAT,

    FOREIGN KEY (HangXe\_Ma) REFERENCES HangXe(MaHang),

    FOREIGN KEY (DongXe\_Ma) REFERENCES DongXe(MaDong),

    FOREIGN KEY (PhienBan\_Ma) REFERENCES PhienBan(MaPhienBan),

    FOREIGN KEY (PhanKhucXe\_Ma) REFERENCES PhanKhucXe(MaPhanKhuc),

    FOREIGN KEY (DongCo\_Ma) REFERENCES DongCo(MaDongCo)

);

INSERT INTO HangXe (MaHang, TenHangXe) VALUES('H001', 'Toyota');

INSERT INTO HangXe (MaHang, TenHangXe) VALUE ('H002', 'Honda');

INSERT INTO HangXe (MaHang, TenHangXe) VALUES ('H003', 'Ford');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D001', 'Camry', 'H001');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES('D002', 'Accord', 'H002');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES('D003', 'Focus', 'H003');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC001', 'V6');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES('DC002', 'V4');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC003', 'EcoBoost');

INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK001', 'Sedan', 'DC001');

INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK002', 'Coupe', 'DC002');

INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK003', 'Hatchback', 'DC003');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc) VALUES ('PB001', 'Camry 2022', 'PK001');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc) VALUES('PB002', 'Accord 2023', 'PK002');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc) VALUES('PB003', 'Focus 2021', 'PK003');

INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan, GiaNiemYet) VALUES

('GPB001', 'PB001', 1000000000);

INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan, GiaNiemYet) VALUES

('GPB002', 'PB002', 900000000);

INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan, GiaNiemYet) VALUES

('GPB003', 'PB003', 800000000);

INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy) VALUES

('HN', 'Hà Nội', 2000000);

INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy) VALUES

('HCM', 'Hồ Chí Minh', 1500000)

INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy) VALUES

('DN', 'Cần Thơ', 1000000);

INSERT INTO ThongSoKyThuat (TSKT\_Ma, MaDongCo, CongSuat, SoCho) VALUES

('TSKT001', 'DC001', 300, 5);

INSERT INTO ThongSoKyThuat (TSKT\_Ma, MaDongCo, CongSuat, SoCho) VALUES

('TSKT002', 'DC002', 200, 5);

INSERT INTO ThongSoKyThuat (TSKT\_Ma, MaDongCo, CongSuat, SoCho) VALUES

('TSKT003', 'DC003', 250, 5);

INSERT INTO Xe (MaXe, HangXe\_Ma, DongXe\_Ma, PhanKhucXe\_Ma, TSKT\_Ma, MaTinhThanh) VALUES  ('X001', 'H001', 'D001', 'PK001', 'TSKT001', 'HN');

INSERT INTO Xe (MaXe, HangXe\_Ma, DongXe\_Ma, PhanKhucXe\_Ma, TSKT\_Ma, MaTinhThanh) VALUES('X002', 'H002', 'D002', 'PK002', 'TSKT002', 'HCM');

INSERT INTO Xe (MaXe, HangXe\_Ma, DongXe\_Ma, PhanKhucXe\_Ma, TSKT\_Ma, MaTinhThanh) VALUES('X003', 'H003', 'D003', 'PK003', 'TSKT003', 'DN');

INSERT INTO GiaXe (HangXe\_Ma, DongXe\_Ma, PhienBan\_Ma, PhanKhucXe\_Ma, DongCo\_Ma, GiaNiemYet) VALUES

('H001', 'D001', 'PB001', 'PK001', 'DC001', 1000000000);

INSERT INTO GiaXe (HangXe\_Ma, DongXe\_Ma, PhienBan\_Ma, PhanKhucXe\_Ma, DongCo\_Ma, GiaNiemYet) VALUES

('H002', 'D002', 'PB002', 'PK002', 'DC002', 900000000);

INSERT INTO GiaXe (HangXe\_Ma, DongXe\_Ma, PhienBan\_Ma, PhanKhucXe\_Ma, DongCo\_Ma, GiaNiemYet) VALUES

('H003', 'D003', 'PB003', 'PK003', 'DC003', 800000000);

2.Viết thủ tục cho phép nhập vào 1 xe mới

CREATE OR REPLACE PROCEDURE ThemXeMoi\_GiaXe (

    p\_HangXe\_Ma VARCHAR2,

    p\_DongXe\_Ma VARCHAR2,

    p\_PhienBan\_Ma VARCHAR2,

    p\_PhanKhucXe\_Ma VARCHAR2,

    p\_DongCo\_Ma VARCHAR2,

    p\_GiaNiemYet FLOAT

)

IS

BEGIN

    INSERT INTO GiaXe (HangXe\_Ma, DongXe\_Ma, PhienBan\_Ma, PhanKhucXe\_Ma, DongCo\_Ma, GiaNiemYet)

    VALUES (p\_HangXe\_Ma, p\_DongXe\_Ma, p\_PhienBan\_Ma, p\_PhanKhucXe\_Ma, p\_DongCo\_Ma, p\_GiaNiemYet);

    COMMIT;

    END;

3. Gọi thủ tục để Nhập liệu như bảng trên

EXECUTE ThemXeMoi\_GiaXe('H001', 'D001', 'PB001', 'PK001', 'DC001', 1000000000);

4.

CREATE OR REPLACE PROCEDURE CapNhatGiaXe (

    p\_MaXe VARCHAR2,

    p\_GiaMoi FLOAT

)

IS

BEGIN

    UPDATE GiaXe

    SET GiaNiemYet = p\_GiaMoi

    WHERE HangXe\_Ma = (

        SELECT HangXe\_Ma

        FROM Xe

        WHERE MaXe = p\_MaXe

    );

    COMMIT;

END;

EXECUTE CapNhatGiaXe('X004', 950000000);

5.

CREATE OR REPLACE FUNCTION LayGiaXeTheoHang (

    p\_TenHangXe VARCHAR2

)

RETURN FLOAT

IS

    v\_Gia FLOAT;

BEGIN

    SELECT G.GiaNiemYet INTO v\_Gia

    FROM GiaXe G

    INNER JOIN Xe X ON G.HangXe\_Ma = X.HangXe\_Ma

    INNER JOIN HangXe H ON X.HangXe\_Ma = H.MaHang

    WHERE H.TenHangXe = p\_TenHangXe;

    RETURN v\_Gia;

END;

SELECT LayGiaXeTheoHang('Toyota') FROM dual;

6.

CREATE OR REPLACE FUNCTION LayLoaiVaHangXeGiaCaoNhat

RETURN VARCHAR2

IS

    v\_LoaiXe VARCHAR2(50);

    v\_HangXe VARCHAR2(50);

BEGIN

    SELECT X.TenDong || ' ' || H.TenHangXe INTO v\_LoaiXe

    FROM Xe X

    INNER JOIN HangXe H ON X.HangXe\_Ma = H.MaHang

    INNER JOIN GiaXe G ON X.HangXe\_Ma = G.HangXe\_Ma

    WHERE G.GiaNiemYet = (SELECT MAX(GiaNiemYet) FROM GiaXe);

    RETURN v\_LoaiXe;

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN NULL;

END;

SELECT LayLoaiVaHangXeGiaCaoNhat() FROM dual;

7.

CREATE OR REPLACE PROCEDURE XoaXe (

    p\_MaXe VARCHAR2

)

IS

BEGIN

    DELETE FROM Xe

    WHERE MaXe = p\_MaXe;

    COMMIT;

END;

EXECUTE XoaXe('X001');

8.

CREATE OR REPLACE FUNCTION TinhGiaLanBanhCanTho (

    p\_MaXe VARCHAR

)

RETURN FLOAT

IS

    v\_GiaLanBanh FLOAT;

    v\_PhiDangKiem FLOAT := 340000;

    v\_BaoHiemTrachNhiemDanSu FLOAT := 437000;

    v\_PhiSuDungDuongBo FLOAT := 1560000;

    v\_PhiCuocBa FLOAT := 65900000;

BEGIN

    SELECT (G.GiaNiemYet + G.PhiCuocBa + XD.PhiDangKy + XD.PhiDangKiEm + XD.PhiSuDungDuongBo + XD.BaoHiemTrachNhiemDanSu + v\_PhiDangKiem)

    INTO v\_GiaLanBanh

    FROM GiaPhienBan GP

    INNER JOIN PhienBan P ON GP.MaPhienBan = P.MaPhienBan

    INNER JOIN Xe X ON P.MaPhanKhuc = X.PhanKhucXe\_Ma

    INNER JOIN NoiDangKy XD ON X.MaTinhThanh = XD.MaTinhThanh

    WHERE X.MaXe = p\_MaXe;

    RETURN v\_GiaLanBanh;

END;

EXECUTE :result := TinhGiaLanBanhCanTho('X001');

9.

CREATE TABLE LichSuCapNhatGiaXe (

    TenXe VARCHAR2(50),

    GiaCu FLOAT,

    GiaMoi FLOAT,

    NgayCapNhat DATE

);

CREATE OR REPLACE TRIGGER CapNhatGiaXe

AFTER UPDATE OF GiaNiemYet ON GiaXe

FOR EACH ROW

DECLARE

    v\_TenXe VARCHAR2(50);

BEGIN

    -- Lấy tên xe từ bảng Xe

    SELECT X.TenXe INTO v\_TenXe

    FROM Xe X

    WHERE X.MaXe = :OLD.MaXe;

    -- Thêm bản ghi vào bảng lưu trữ thay đổi

    INSERT INTO LichSuCapNhatGiaXe (TenXe, GiaCu, GiaMoi, NgayCapNhat)

    VALUES (v\_TenXe, :OLD.GiaNiemYet, :NEW.GiaNiemYet, SYSDATE);

END;

-- Cập nhật giá mới cho xe có mã 'X002' từ 1 tỷ đến 5 tỷ

UPDATE GiaXe

SET GiaNiemYet = 5000000000

WHERE MaXe = 'X002';

SELECT \* FROM LichSuCapNhatGiaXe;

10.

-- Bước 1: Tạo bảng và trigger

CREATE TABLE LichSuXoaXe (

    MaXe VARCHAR2(15),

    TenXe VARCHAR2(50),

    NgayXoa DATE

);

CREATE OR REPLACE TRIGGER XoaXe\_Trigger

AFTER DELETE ON Xe

FOR EACH ROW

BEGIN

    -- Thêm bản ghi vào bảng lưu trữ thay đổi

    INSERT INTO LichSuXoaXe (MaXe, TenXe, NgayXoa)

    VALUES (:OLD.MaXe, :OLD.TenXe, SYSDATE);

END;

-- Bước 2: Thực hiện câu lệnh DELETE để xóa một xe

DELETE FROM Xe

WHERE MaXe = 'X001';

-- Bước 3: Kiểm tra bảng LichSuXoaXe để xem liệu bản ghi đã được thêm vào hay không

SELECT \* FROM LichSuXoaXe;

**11. Tạo các người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..)**

-- Tạo người dùng mới

CREATE USER Quantri IDENTIFIED BY quanti;

-- Cấp quyền quản trị viên cho người dùng Quantri

GRANT DBA TO Quantri;

-- Tạo người dùng mới cho nhân viên bán xe

CREATE USER NhanVienBanXe IDENTIFIED BY NVbanxe;

-- Cấp quyền chỉ đọc cho người dùng NhanVienBanXe

GRANT SELECT ON Xe TO NhanVienBanXe;

GRANT SELECT ON HangXe TO NhanVienBanXe;

GRANT SELECT ON DongXe TO NhanVienBanXe;

GRANT SELECT ON PhanKhucXe TO NhanVienBanXe;

GRANT SELECT ON PhienBan TO NhanVienBanXe;

GRANT SELECT ON GiaXe TO NhanVienBanXe;

-- Tạo người dùng mới cho kế toán

CREATE USER KeToan IDENTIFIED BY ketoan;

-- Cấp quyền chỉ đọc và cập nhật cho người dùng KeToan

GRANT SELECT, UPDATE ON GiaXe TO KeToan;

-- Tạo người dùng mới cho quản lý bảo dưỡng

CREATE USER QuanLyBaoDuong IDENTIFIED BY qlbaoduong;

-- Cấp quyền chỉ đọc và cập nhật cho người dùng QuanLyBaoDuong

GRANT SELECT, UPDATE ON Xe TO QuanLyBaoDuong;

GRANT SELECT, UPDATE ON ThongSoKyThuat TO QuanLyBaoDuong;